Control of Autonomous Underwater Vehicle (AUV)

Students:
Shahid Ahamed Hasib
Faith Oluwasegun Mustapha
Hyejoo Kwon
Mohamed Magdy Atta
Ahmed Borchani

Course Title:
Linear Multivariable Control

Instructor:
DHAISNE Aurélien

Submitted on October 23, 2025

UNIVERSITÉ DE TOULON

MIR MARINE & MARITIME INTELLIGENT ROBOTICS

Contents

1 Introduction **2**
- 1.1 Characteristics of the Underwater Environments . . . 2
- 1.2 Autonomous Underwater Vehicle . . . 2
- 1.3 Need for an Improved Control Systems . . . 2
- 1.4 Aim and Objectives . . . 3

2 Identification **3**
- 2.1 Speed step . . . 3
- 2.2 Pitch step . . . 4
- 2.3 Dieudonnée . . . 5

3 Linearization: A and B and Equilibrium points **6**
- 3.1 Selecting the State X . . . 6
- 3.2 Finding the equilibrium points . . . 7
- 3.3 Linearization of the equations . . . 8
- 3.4 Caculation of the matrices A and B . . . 8

4 LQR control **9**
- 4.1 MATLAB Implementation . . . 9
- 4.2 Selection of Q and R Matrices . . . 10
- 4.3 Computation of Feedback Gain Matrix K_d . . . 10
- 4.4 Limitations and Benefits of LQR . . . 10
  - 4.4.1 Limitations of LQR . . . 10
  - 4.4.2 Benefits of LQR . . . 11

5 Simulation and Results **11**

6 Conclusion **15**

1 Introduction

1.1 Characteristics of the Underwater Environments

Exploring and operating crafts in the underwater environment comes with unique challenges. Characterized by high pressures, limited visibility, dynamic currents, and unpredictable thermal gradients, this environment poses significant challenges for both human divers and robotic systems. These conditions create a complex and dynamic operational landscape, where factors such as as buoyancy changes, wave-induced forces, and interactions with marine life introduce inherent non-linearities and disturbances. In addition, the acoustic properties of water, which are key to sonar and communication, impose limitations, such as attenuation and distortion of signals. Therefore, there is a need to precisely control vehicle stability for effective and safe operations.

1.2 Autonomous Underwater Vehicle

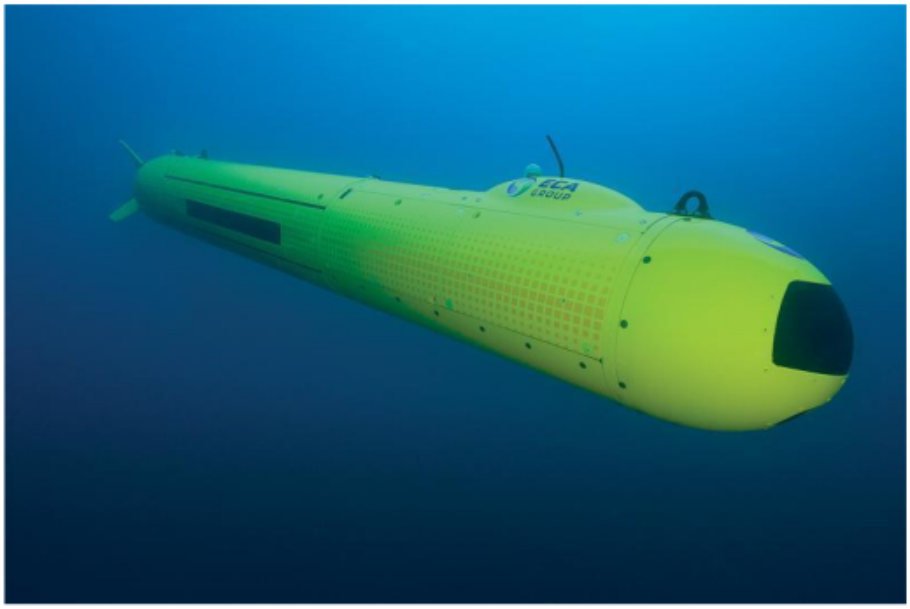

Figure 1: An A18M AUV

Autonomous Underwater Vehicles (AUVs) have become important tools for conducting tasks ranging from scientific exploration to naval defense in these challenging environments. Their ability to autonomously navigate and gather data without human intervention has brought about significant progress in fields such as marine biology, geoscience, and naval defense. Specifically, AUVs like the A18M developed by Exail Robotics are equipped with advanced sonar systems designed for detecting underwater threats -mines. These mines pose a significant risk to military vessels and submarines due to their cost-efficiency and effectiveness in destroying vessels and submarines. For a task like this, stability across all planes, vertical, horizontal, and roll, surge, sway, and heave is critical to ensure no mine is omitted and the mission is successfully executed.

1.3 Need for an Improved Control Systems

The effectiveness of AUVs depends on robust control systems that can effectively address the non-linearities and dynamic nature of the underwater environment. Proportional Integral Derivative (PID) controllers previously used in AUVs lacks robustness and stability. Their performance in highly non-linear underwater environments is often not optimal. The introduction

of new, high-performance sensors further increases the need for more advanced control strategies. The stability and robustness demanded by these sensors are beyond what PID controllers can provide, necessitating the development of a new optimal control method to improve performance, reduce energy consumption, and ultimately improve the success rates of underwater mines detection.

Linear Quadratic Regulator (LQR) controllers offer a practical and efficient solution to these challenges. By minimizing a quadratic cost function that balances control effort and system state deviations, LQR controllers achieve enhanced stability and precision. This makes them particularly well-suited for applications in the vertical plane, where precise depth control and minimal pitch oscillations are critical.

1.4 Aim and Objectives

The aim of this practical work is to design and implement an optimal control command law for the vertical plane motion of an AUV. This involves refining the current control architecture of PID-based systems to a more advanced Linear Quadratic Regulator (LQR) approach.

This report outlines the process of developing an LQR-based control law for the vertical plane of an AUV, focusing on the depth command. The objectives include:

1. The identification of missing hydrodynamic parameters.
2. Linearization of the AUV's nonlinear model.
3. Tuning of the LQR controller to achieve low final error by aligning with the desired performance metrics:
   (a) Low pitch oscillations
   (b) Low lift speed variation
   (c) Minimal control overshoot

2 Identification

As a first step towards controlling the AUV, its model needs to be derived. The missing parameters that are calculated in this project are the damping parameters. The rest is already provided.

2.1 Speed step

When moving in the in the surge motion with pure speed steps u, we can use the recorded data to calculate the parameter K_{sh}. According to the provided equations, the hydrodynamic damping force in the forward plane is:

$$F_x = \frac{1}{2} \cdot \rho \cdot S_{\text{ref}} \cdot C_{X_0} \cdot u \cdot |u|$$

where:

$$C_{X_0} = K_{\text{sh}} \cdot \frac{0.075}{(\log_{10}(R_e) - 2)^2}$$

and:

$$R_e = \frac{u \cdot L_{\text{ref}}}{\nu}$$

ν is the cinematic viscosity of sea water, it's a constant. u is the AUV speed in the surge direction, like F_x, several data points are provided from previous missions. To estimate the

parameter K_{sh}, least square method is used. The latter tries to identify the parameter that minimizes the error between the estimate and the actual data. The following result is achieved:

$$K_{\text{sh}} = -43.986$$

The least square method works by trying to minimize the difference between the observed data and the model. For each data point i, this difference (called the residual) is computed and summed, and it has to be minimized:

$$min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where y_i is the actual data point, and $\hat{y}_i$ is the estimate based on the chosen value for the parameter we are trying to calculate.

2.2 Pitch step

The next parameters to identify are:

- C_{Zuw}, C_{Zuq}: damping coefficients of force F_z by the product of $u \cdot w$ and $u \cdot q$ respectively.
- C_{Muw}, and C_{Muq}: Damping coefficient of moment M_y by the product of $u \cdot w$ and $u \cdot q$ respectively.

Based on the final equations, the hydrodynamic damping parameters follow:

$$F_z = C_{Z0}|u_0|u_0 + C_{Zuw}u_0 w + C_{Zuq}L_{\text{ref}}u_0 q$$

and

$$M_y = L_{\text{ref}}C_{M0}|u_0|u_0 + C_{Muw}L_{\text{ref}}u_0 w + L_{\text{ref}}^2 C_{Muq}u_0 q$$

These equations are derived with the hypothesis that $u = u_0$, the actual u data is used instead of the constant u_0.

To be able to ignore the rest of the speeds and focus on q (the pitch angular velocity) and w (the heave velocity), we use data of a mission where only pitch steps were taken. Then, least-square method uses this data on the equations to approximate the parameters values:

$$C_{Zuw} = -3.158$$

$$C_{Zuq} = -1.939$$

Using these estimated parameters, we compute F_z and compare it to the actual data, the results are almost identical:

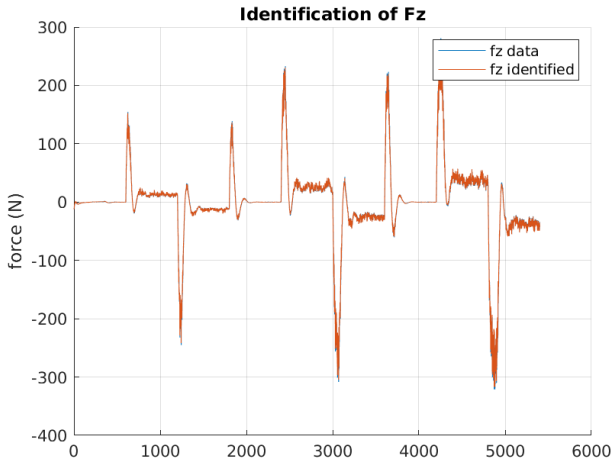

Figure 2: F_z actual and estimated data

As for C_{Muw} and C_{Muq}, the following results are computed:

$$C_{Muw} = 0.913$$

$$C_{Muq} = -0.867$$

When plugging in these values to determine M_y, the following graphs are obtained:

Figure 3: M_y actual and estimated data

The approximation is not as accurate as for F_z but it still provides parameters that follow the actual data. After running the least-square command on MATLAB (lsqr) for each of the parameters, the final residual after convergence is provided. As for K_{sh} it is 0.026, for $C_{Zuw} and C_{Zuq}$ it is 0.14, while for C_{Muw} and C_{Muq} it is 0.31. This explains the difference in results when comparing to the actual data.

2.3 Dieudonnée

This type of mission generates data about the AUV behavior in the horizontal plane, that we use for identifying parameters C_Y (damping coefficients of force F_y) and C_N (damping coefficients of moment M_z). In this mission, the four physical helms are used in a way to create virtual helms steps in the horizontal plane. Thus, we get significant values for the sway v and the yaw r, while the rest of speeds are negligible.

For identifying parameters C_Y and C_N, the use of least squares is not enough to find a solution. This is because there are several local minima and least squares would provide the first that it meets, which can be far from the optimal value.

Instead, we use fmincon, which minimizes a scalar function given some constraints. The symmetry in the AUV shape provides the constraints we need to find the parameters:

- $C_{Yuv} \sim C_{Zuw}\,(\pm 20\%)$
- $C_{Yur} \sim -C_{Zuq}\,(\pm 2.5\%)$
- $C_{Nuv} \sim -C_{Muw}\,(\pm 10\%)$
- $C_{Nur} \sim C_{Muq}\,(\pm 10\%)$

These constraints help avoid local minima that do not yield realistic or accurate estimations, and they narrow the search space for values that minimize the difference between the predicted and actual data.

After using MATLAB to solve fmincon, the following results are obtained:

$$C_{Yuv} = -3.790$$

$$C_{Yur} = 1.945$$

$$C_{Nuv} = -0.822$$

$$C_{Nur} = -0.948$$

The simulated results based on the estimated parameters are very close to the actual data:

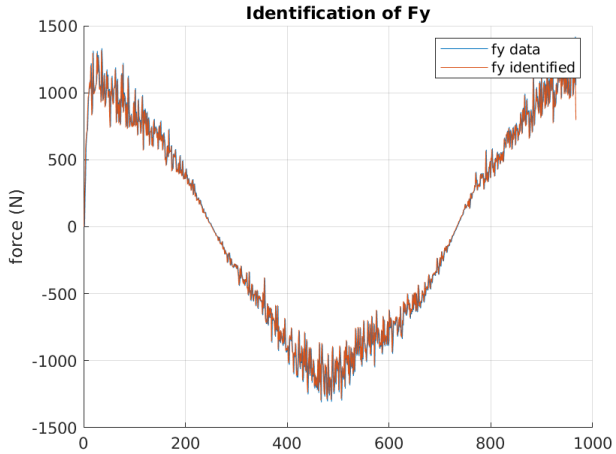

Figure 4: F_y actual and estimated data

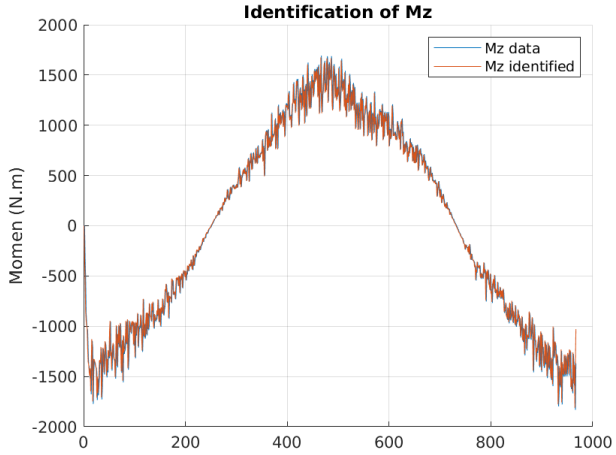

Figure 5: M_z actual and estimated data

3 Linearization: A and B and Equilibrium points

In this section, we explain how to calculate the equilibrium points and determine the A and B matrices using the reference equations below. We break down the steps to make it easy to understand how these key components are derived for the system.

3.1 Selecting the State X

The first step is to carefully select the state vector X, which represents the key variables describing the system's behavior.

For this system, based on the final equations, and in order to control the depth command, the chosen state vector is defined as:

$$X = \begin{bmatrix} q & w & z & \theta & \int z \end{bmatrix}$$

- q: Angular velocity about y axis, describing the rotation along the pitch angle.
- w: Heave velocity, representing the movement along the verticle axis.
- z: Depth, representing the vertical position.
- θ: Pitch angle, showing the orientation of the AUV.
- $\int z$: Integral of depth error: We introduce it to eliminate steady state error since it is a requirement.

3.2 Finding the equilibrium points

The second step is to calculate the equilibrium points of the system. To achieve this, the dynamic and kinetic equations of the system described below are used.

At the equilibrium, we need to have a zero depth error and zero pitch velocity. In addition, the depth would be at the requested value, and $\dot{z}$ should be zero. Therefore, the equilibrium points for the state vector X can be simplified as follows:

$$\bar{w} = u_0 \tan(\bar{\theta}), \quad \bar{q} = 0, \quad \bar{z} = z_{\text{command}}, \quad \int \bar{z} = 0.$$

Thus, we only need to solve for the equilibrium values of the control input (B$\bar{\text{A}}$R) and the pitch angle ($\bar{\theta}$). The reference equations used for this process are:

$$\begin{bmatrix} \dot{u}_0 \\ \dot{w} \\ \dot{q} \end{bmatrix} = \underbrace{M^{-1}}_{\text{inverse of the mass matrix}} \Bigg\{ \underbrace{\frac{1}{2}\rho S_{\text{ref}} \begin{bmatrix} \frac{1}{4}C_X(A^2 + BAR^2 + G^2 + D^2)u_0|u_0| \\ C_Z BAR u_0|u_0| \\ L_{\text{ref}} C_M BAR u_0|u_0| \end{bmatrix}}_{\text{Actuators}}$$

$$+ \underbrace{\begin{bmatrix} 0 & 0 & -mw \\ 0 & 0 & m(z_g q + u_0) \\ mw & -m(z_g q + u_0) & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ w \\ q \end{bmatrix} \Bigg\}}_{\text{Coriolis/Centerpete Vehicle and Added Water Mass}}$$

$$+ \underbrace{\frac{1}{2}\rho S_{\text{ref}} \begin{bmatrix} C_{X0}|u_0| & 0 & 0 \\ C_{Z0}|u_0| & C_{Zww}u_0 & C_{Zuq}L_{\text{ref}}u_0 \\ L_{\text{ref}}C_{M0}|u_0| & C_{Mww}L_{\text{ref}}u_0 & L_{\text{ref}}^2 C_{Muq}u_0 \end{bmatrix} \begin{bmatrix} u_0 \\ w \\ q \end{bmatrix}}_{\text{Hydrodynamic Damping}} + \underbrace{\begin{bmatrix} 0 \\ (W-B)\cos(\theta) \\ -(z_g W - z_b B)\sin(\theta) \end{bmatrix}}_{\text{Hydrostatic}},$$

$$\begin{bmatrix} \dot{x} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ w \\ q \end{bmatrix}.$$

From those equations, we derived the equilibrium values for ($\bar{\theta}$) and (B$\bar{\text{A}}$R). The symbolic equations for $\bar{\theta}$ and B$\bar{\text{A}}$R are defined as follows:

```
%equilibrium point at theta0 and BAR0
theta_0 = (-(Det_33 * 0.5 * rho * Sref * CZ * u0 * abs(u0)) * (Det_55 * 0.5 * rho * Sref * Lref * CM0 * abs(u0) * u0) + ...
           (Det_33 * 0.5 * rho * Sref * CZ0 * abs(u0) * u0) * (Det_55 * 0.5 * rho * Sref * Lref * CM * u0 * abs(u0)) + ...
           (Det_33 * (W - B)) * (Det_55 * 0.5 * rho * Sref * Lref * CM * u0 * abs(u0))) / ...
          ((Det_33 * 0.5 * rho * Sref * CZ * u0 * abs(u0)) * (Det_55 * 0.5 * rho * Sref * CMuw * Lref * u0) * u0 + ...
           (Det_33 * 0.5 * rho * Sref * CZ * u0 * abs(u0)) * (-Det_55 * (zg * W - zb * B)) - ...
           (Det_33 * 0.5 * rho * Sref * CZuw * u0) * (Det_55 * 0.5 * rho * Sref * Lref * CM * u0 * abs(u0)) * u0);

% Calculate BAR_0 directly
BAR_0 = (-(Det_55 * 0.5 * rho * Sref * CMuw * Lref * u0) * u0 * theta_0 - ...
         (Det_55 * 0.5 * rho * Sref * Lref * CM0 * abs(u0) * u0) - ...
         (-Det_55 * (zg * W - zb * B)) * theta_0) / ...
        (Det_55 * 0.5 * rho * Sref * Lref * CM * u0 * abs(u0));
```

Figure 6: The results of the equilibrium points ($\bar{\theta}$) and (B$\bar{\text{A}}$R).

The final results of $\bar{\theta}$ and B$\bar{\text{A}}$R are calculated as:

$$\text{B}\bar{\text{A}}\text{R} = -0.082$$
$$\bar{\theta} = 0.007$$

3.3 Linearization of the equations

In order to compute A and B matrices, the previously defined equations were rearranged to derive the dynamic expressions for $\dot{w}, \dot{q}, \dot{\theta}, \dot{z}, \int \dot{z}$. These expressions form the foundation for calculating the linearized equations of the system.

Contribution	**Equation**
Actuator forces (heave, pitch)	$F_w = \frac{1}{2}\rho S_{\text{ref}} C_Z u_0 \lvert u_0 \rvert \text{BAR},$ $F_q = \frac{1}{2}\rho S_{\text{ref}} L_{\text{ref}} C_M u_0 \lvert u_0 \rvert \text{BAR}$
Coriolis and added mass forces/moments	$C_w = m(z_g q + u_0) q,$ $C_q = m w u_0 - m(z_g q + u_0) w$
Hydrodynamic damping	$H_w = \frac{1}{2}\rho S_{\text{ref}} \big(C_{Z0} \lvert u_0 \rvert u_0 + C_{Zuw} u_0 w + C_{Zuq} L_{\text{ref}} u_0 q\big),$ $H_q = \frac{1}{2}\rho S_{\text{ref}} \big(L_{\text{ref}} C_{M0} \lvert u_0 \rvert u_0 + C_{Muw} L_{\text{ref}} u_0 w + L_{\text{ref}}^2 C_{Muq} u_0 q\big)$
Hydrostatic forces/ moments	$G_w = (W - B)\cos(\theta),$ $G_q = -(z_g W - z_b B)\sin(\theta)$

$$\begin{aligned}
\cdot\ \dot{w} &= \text{Det}_{33} \cdot \big(F_w + C_w + H_w + G_w\big) \\
\cdot\ \dot{q} &= \text{Det}_{55} \cdot \big(F_q + C_q + H_q + G_q\big) \\
\cdot\ \dot{\theta} &= q \\
\cdot\ \dot{z} &= -\sin(\theta) \cdot u_0 + \cos(\theta) \cdot w \\
\cdot\ \int \dot{z} &= z.
\end{aligned}$$

3.4 Caculation of the matrices A and B

To compute the Jacobian matrix A, we take the partial derivatives of each dynamic equation $(\dot{w}, \dot{q}, \dot{\theta}, \dot{z}, \int \dot{z})$ with respect to state X defined above. On the other hand, the B matrix is computed by taking the partial derivatives of the dynamic equations with respect to BAR. The general form of the A and B matrix are given as:

$$A = \begin{bmatrix} \frac{\partial \dot{w}}{\partial w} & \frac{\partial \dot{w}}{\partial q} & \frac{\partial \dot{w}}{\partial \theta} & \frac{\partial \dot{w}}{\partial z} & \frac{\partial \dot{w}}{\partial \int z} \\ \frac{\partial \dot{q}}{\partial w} & \frac{\partial \dot{q}}{\partial q} & \frac{\partial \dot{q}}{\partial \theta} & \frac{\partial \dot{q}}{\partial z} & \frac{\partial \dot{q}}{\partial \int z} \\ \frac{\partial \dot{\theta}}{\partial w} & \frac{\partial \dot{\theta}}{\partial q} & \frac{\partial \dot{\theta}}{\partial \theta} & \frac{\partial \dot{\theta}}{\partial z} & \frac{\partial \dot{\theta}}{\partial \int z} \\ \frac{\partial \dot{z}}{\partial w} & \frac{\partial \dot{z}}{\partial q} & \frac{\partial \dot{z}}{\partial \theta} & \frac{\partial \dot{z}}{\partial z} & \frac{\partial \dot{z}}{\partial \int z} \\ \frac{\partial \int \dot{z}}{\partial w} & \frac{\partial \int \dot{z}}{\partial q} & \frac{\partial \int \dot{z}}{\partial \theta} & \frac{\partial \int \dot{z}}{\partial z} & \frac{\partial \int \dot{z}}{\partial \int z} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{\partial \dot{w}}{\partial \mathrm{BAR}} \\ \frac{\partial \dot{q}}{\partial \mathrm{BAR}} \\ \frac{\partial \dot{\theta}}{\partial \mathrm{BAR}} \\ \frac{\partial \dot{z}}{\partial \mathrm{BAR}} \\ \frac{\partial \int \dot{z}}{\partial \mathrm{BAR}} \end{bmatrix}.$$

The symbolic results of the matrix A and B are defined as:

$$A = \begin{bmatrix} \mathrm{Det}_{55}(0.5\rho S_{\mathrm{ref}} L^2 C_{M_{uq}} u_0 - m z_g w) & \mathrm{Det}_{55}(-m z_3 q + 0.5\rho S_{\mathrm{ref}} C_{M_{uv}} L_{\mathrm{ref}} u_0) & 0 & -\mathrm{Det}_{55}(z_g W - z_p B)\cos(\theta_0) & 0 \\ \mathrm{Det}_{33}(2 m z_3 q + m u_0 + 0.5\rho S_{\mathrm{ref}} C_{Z_{uq}} L_{\mathrm{ref}} u_0) & \mathrm{Det}_{33}(0.5\rho S_{\mathrm{ref}} C_{Z_{uv}} u_0) & 0 & -\mathrm{Det}_{33}(W - B)\sin(\theta_0) & 0 \\ 0 & \cos(\theta_0) & 0 & -u_0 \cos(\theta_0) - w \sin(\theta_0) & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \mathrm{Det}_{55} \cdot 0.5\rho S_{\mathrm{ref}} L_{\mathrm{ref}} C_M u_0 |u_0| \\ \mathrm{Det}_{33} \cdot 0.5\rho S_{\mathrm{ref}} C_Z u_0 |u_0| \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The final results of the matrix A and B are calculated as:

$$A = \begin{bmatrix} -1.4158 & 0.2982 & 0 & -0.0391 & 0 \\ -0.7978 & -0.5469 & 0 & 4.9225 \times 10^{-6} & 0 \\ 0 & 0.9999 & 0 & -2.5000 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -0.1387 \\ -0.1775 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

4 LQR control

The Linear-Quadratic Regulator (LQR) is a robust technique for designing feedback control systems. It operates by calculating an optimal feedback gain matrix K, which determines how the control input u should be adjusted based on the current state of the system y.

The primary objective of the LQR method is to minimize a cost function J, which represents a trade-off between the squared errors of the system's desired and actual states and the effort required to control the system (i.e., the squared control inputs). Mathematically, the cost function is expressed as:

$$J = \int_0^\infty \left(y^T Q y + u^T R u \right) dt$$

Here:

- Q: A weighting matrix penalizing deviations from the desired state.
- R: A weighting scalar or matrix penalizing the magnitude of the control inputs.

4.1 MATLAB Implementation

In MATLAB, the `lqrd` command is utilized to compute the discrete full-state feedback regulation gain K_d for a continuous-time system. This approach enables the effective regulation of discrete-time systems derived from their continuous-time counterparts.

The K matrix is the **feedback gain matrix** in a Linear Quadratic Regulator (LQR) system. It determines how the state variables are fed back to stabilize the system and optimize the desired performance.

The expression $Q = C^\top QC$ adjusts the weighting matrix Q based on the system's **output matrix** C. The reasons behind this are as follows:

- Q is a weighting matrix that penalizes state deviations from the desired values.
- C often maps the system state x to the measurable outputs.
- Pre-multiplying by $C^\top$ and post-multiplying by C ensures that the penalization (weighting) of state deviations aligns with the system outputs, focusing on measurable states or outputs instead of internal states that might not be directly observable.

This adjustment ensures that Q incorporates the structure of C to weight the outputs more meaningfully.

4.2 Selection of Q and R Matrices

For this implementation, the Q and R matrices were chosen as follows:

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix}$$

$$R = 1e6$$

After experimenting, with trial and error method we found those optimal values. Choosing the right Q and R value is essential.

- Q: Focuses on penalizing certain state variables more than others. For instance:
  - 1×10^{-6}: A very low weight for the first state, meaning deviations here are not heavily penalized.
  - 100: A higher weight for a specific state, meaning it's crucial to keep this state close to its target.
- R: Penalizes control effort. A high R (like 10^9) implies that applying control effort is very costly, so the system will use minimal actuation.

4.3 Computation of Feedback Gain Matrix K_d

At each time step, the feedback gain matrix K_d is calculated using MATLAB's `lqrd` command. This ensures the system remains stable and adheres to the desired performance criteria. By systematically updating K_d, the control system achieves optimal regulation under discrete-time conditions.

4.4 Limitations and Benefits of LQR

4.4.1 Limitations of LQR

1. **Linear Systems**: LQR is designed for linear systems, limiting its direct applicability to nonlinear systems without prior linearization.
2. **Robustness**: LQR does not inherently address robustness to uncertainties. Extensions like H_2 or H_∞ methods are needed for robust control.

3. **Equilibrium Point Model**: LQR is designed to regulate systems around an equilibrium point, making it less effective for systems with highly dynamic or non-equilibrium behaviors.

4. **Saturation and Constraints**: LQR does not handle input or state constraints, requiring additional techniques (like predictive control) for such scenarios.

5. **Predictive Control (MPC) Alternatives**: Model Predictive Control (MPC) offers more flexibility by incorporating constraints and adapting to changing system dynamics, which LQR lacks.

6. **Model Dependency**: The performance of LQR heavily depends on an accurate system model; any discrepancies can degrade its performance.

4.4.2 Benefits of LQR

1. **Model-Based Design**: LQR utilizes a detailed model of the system, ensuring optimal performance based on the given dynamics.

2. **Gain and Phase Margins**: LQR provides stability margins such as gain and phase margins, typically around 45°, ensuring robust operation under small perturbations.

3. **Robustness**: While not inherently robust, LQR achieves a certain degree of robustness when tuned appropriately, especially for small modeling errors.

5 Simulation and Results

In this section, we discuss and analyze the results from the simulation performed in MATLAB and SIMULINK software. The purpose of the simulation is to validate the tuned optimal control law by applying it to the A18M AUV in a simulated scenario for mine detection. In our work we focused on controlling the depth in the vertical plane during the vehicle movement. To validate our controller, a number of macroscopic behaviors are studied in the following paragraphs.

First of all, we check the XY plane movements of the AUV. The path generated by the A18M AUV during its mission is shown below:

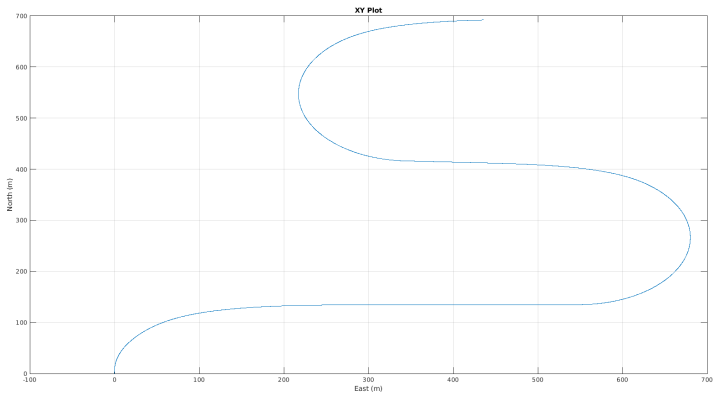

Figure 7: XY Trajectory

As seen in figure [8], the trajectory executed to detect the mines is correct. The vehicle moves in a straight lines separated by some distance between them, and there are smooth turns between the straight lines.

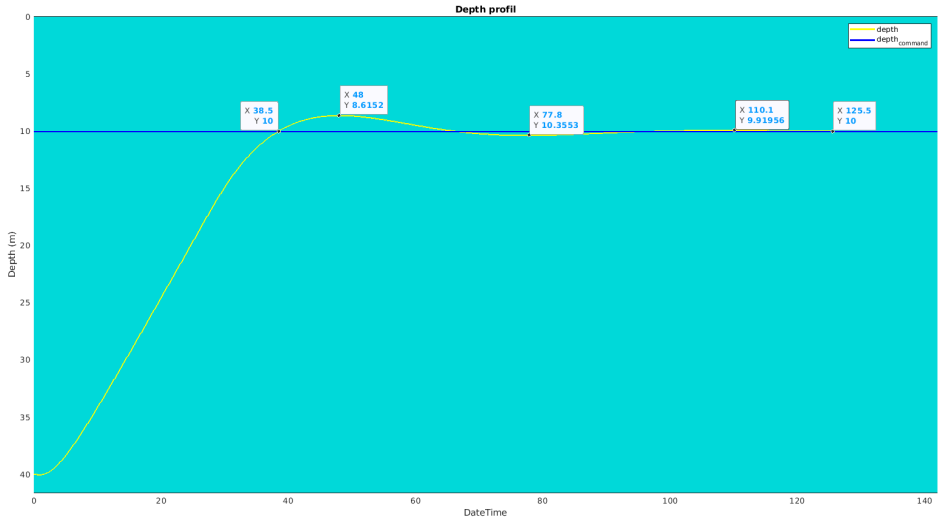

Figure 8: Z Trajectory

Secondly, we validate the vertical movements of the AUV in the following plot:

As seen in figure [8], the AUV responds to a depth command using the LQR command. The steady state was reached after around 125.5 seconds. The goal was to reach 10 meters depth. The maximum overshoot by around 1.4 meters happened at 8.6 meters depth which is less than the maximum allowed overshoot (2 meters). The final error is approximately zero and there is no oscillation around the command, which is a requirement also. Given these results, the performance is acceptable.

Thirdly, we check the behavior of the angle around y-axis in the following plot:

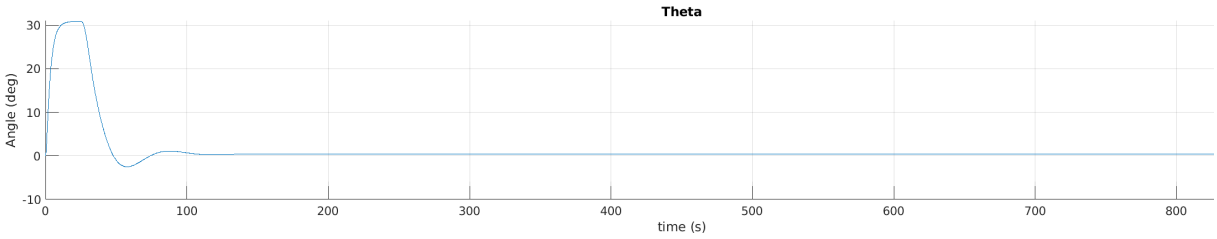

Figure 9: Pitch angle

As seen in figure [9], the pitch angle have a positive values to make the vehicle goes up from the initial depth of 40 meters to the commanded depth at 10 meters. After reaching the desired depth after around 125 seconds, the pitch angle starts to stabilize near the zero value.

To adjust the pitch angle to the desired positive values, the AUV rear helms are adjusted and the values are shown in the following BAR plot:

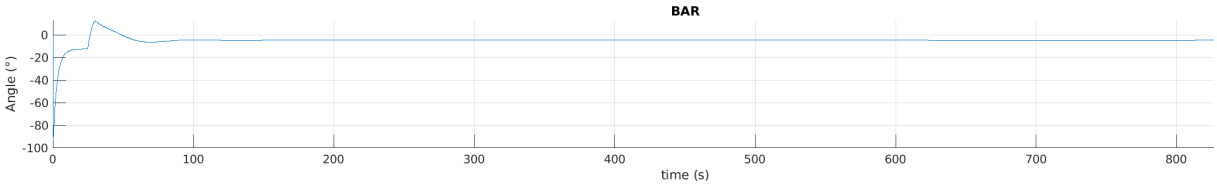

Figure 10: BAR

As seen in figure [10], the angle is adjusted to be negative at the beginning to make the

vehicle goes up in the vertical plane, then the angle stabilize near the zero after reaching the desired depth.

The rear motor RPM speed is approximately constant through the motion to ensure smooth and stable movements as shown in the figure [11].

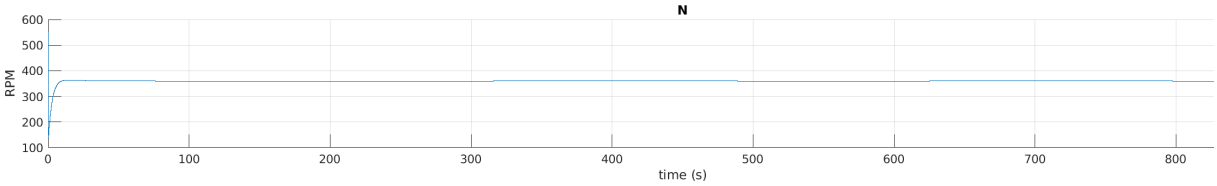

Figure 11: Motor Speed RPM

Next, we analyze the linear speed along the z-axis using the figure [12].

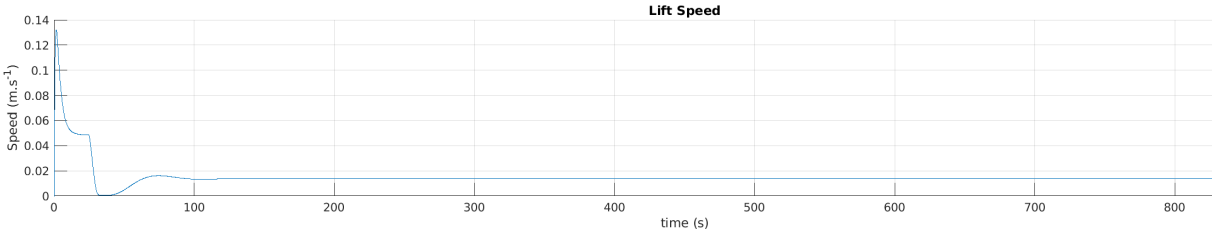

Figure 12: Lift Speed

As seen in figure [12], we observe a positive lift speed in the beginning of the simulation. This positive values contribute to the upward motion to reach the desired depth. Also, one of the requirements was to not having an oscillating lift speed along with a low value of it, and this was achieved as shown in the figure.

In addition to the lift speed, we observed a speed along the x-axis (2 m/s) as shown in figure [13] below. This longitudinal speed contributed to the depth control as well and helped the AUV reaching the desired depth while moving in the XY plane

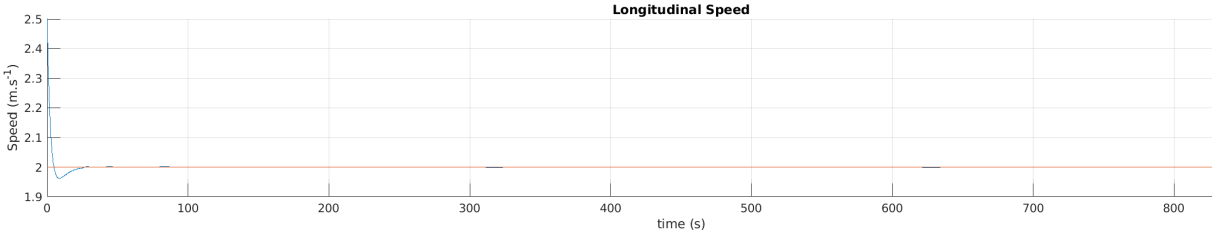

Figure 13: Longitudinal Speed

Finally, we analyze the behavior of the angular speed along the y-axis using the figure [14].

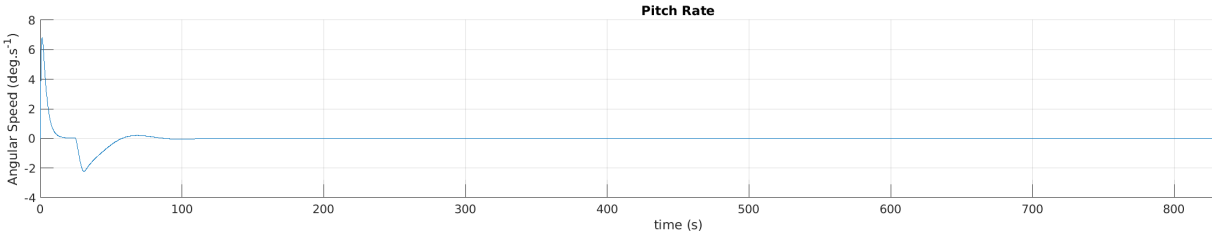

Figure 14: Pitch angular Speed

As seen in figure [14], we observe a positive values of pitch rate in the beginning, which aligns with the positive pitch angle values generated as well. This behavior will result in going upward in the z-direction while moving in XY plane as required. After reaching the desired depth, the pitch rate values stabilizes around the zero, which is also a requirement to not having too many pitch oscillation and having low q values, and this can be seen in the figure [14].

6 Conclusion

By addressing the challenges of non-linearity and disturbances inherent in underwater environments, this study has achieved improved depth control, and stability which ensures that the sonar maintains its optimal operating conditions, achieving precise alignment and consistent scanning patterns, which are essential for detecting and classifying underwater mines.

The simulation results confirmed that the LQR controller met all the pre-defined performance criteria. The depth command was executed with a steady-state error close to zero, and the overshoot of 1.4 meters was well within the acceptable limit of 2 meters. Moreover, the absence of oscillations around the commanded depth underscores the precision of the control system. This stability is crucial for the effective operation of the sonar sensor, ensuring high-quality data acquisition and reliable detection of mines.

The pitch angle and pitch angular velocity, key contributors to the vertical motion, showed smooth transitions and stabilized near zero after reaching the desired depth. This behavior highlights the robustness of the control strategy in minimizing unnecessary energy expenditure and maintaining operational efficiency. The lift speed remained low and free of oscillations, further demonstrating the effectiveness of the controller in achieving steady vertical movement.

Actuator performance also exhibited predictable and smooth behavior, with rear helms and motor speeds maintaining consistent adjustments to support the vehicle's motion. This ensures a longer operational life of the actuators while sustaining mission-critical stability. The longitudinal speed of 2 m/s played a supportive role in maintaining depth control while adhering to the vehicle's trajectory in the XY plane, confirming the integrated performance of all subsystems.

While the results are promising, the limitations of the LQR approach, including its dependence on accurate linearization and lack of inherent robustness to large uncertainties, should be noted. Future work could explore hybrid control systems or adaptive methods to further improve the vehicle's performance in more complex and dynamic environments.